Improving Content Availability in the I2P Anonymous File-Sharing Environment

Juan Pablo Timpanaro, Isabelle Chrisment, Olivier Festor

Improving Content Availability in the I2P Anonymous File-Sharing Environment

Juan Pablo Timpanaro, Isabelle Chrisment*, Olivier Festor

INRIA Nancy-Grand Est, France
*LORIA - ESIAL, Université de Lorraine
Email: {juanpablo.timpanaro, olivier.festor}@inria.fr
Email: {isabelle.chrisment}@loria.fr

Abstract. Anonymous communication has gained more and more interest from Internet users as privacy and anonymity problems have emerged. Dedicated anonymous networks such as Freenet and I2P allow anonymous file-sharing among users. However, one major problem with anonymous file-sharing networks is that the available content is highly reduced, mostly with outdated files, and non-anonymous networks, such as the BitTorrent network, are still the major source of content: we show that in a 30-days period, 21648 new torrents were introduced in the BitTorrent community, whilst only 236 were introduced in the anonymous I2P network, for four different categories of content.
Therefore, how can a user of these anonymous networks access this varied and non-anonymous content without compromising its anonymity? In this paper, we improve content availability in an anonymous environment by proposing the first internetwork model allowing anonymous users to access and share content in large public communities while remaining anonymous. We show that our approach can efficiently interconnect I2P users and public BitTorrent swarms without affecting their anonymity nor their performance. Our model is fully implemented and freely usable.

1 Introduction

Peer-to-peer file-sharing has always been one of the major sources of the Internet traffic, since its early beginnings in 2000. It has been moving from semi-central approaches (eDonkey2000, for example), to semi-decentralized approaches (Kazaa, for instance) to fully decentralized file-sharing architectures (like the KAD network). Nowadays, it is still a major activity within the Internet, despite being constantly supervised by governmental institutions, law-enforcement agencies and movie-maker agencies, among others, mostly due to copyrighted file-sharing. Moreover, the recent legal actions against Megaupload, a major file-sharing website, and the repercussions on the rest of these file-sharing websites will definitely increase P2P traffic.

Both privacy and anonymity definitions have been gaining attention in the Internet. More and more users are realizing the importance of maintaining a certain degree of anonymity when accessing the Internet so as to keep their online

ideas and their real identities separated. Within the P2P world, anonymous file-sharing is usually linked with illegal or copyrighted downloads; however, maintaining a user identity hidden is important to avoid censorship by certain institutions, avoiding file-sharing profiling through data mining or retaliation against the uploading/downloading of unofficial leaks, among others.

Despite the wide range of anonymous file-sharing options, one of the problems is that the biggest sharing communities are still public. Let's consider the BitTorrent community, which is one of the biggest content distribution communities. Which options do BitTorrent users have to become anonymous whilst downloading? And regarding users that already formed part of an anonymous network, how can these users access BitTorrent content from their anonymous networks?

In the first case, many BitTorrent users are routing their traffic through paid VPNs or dedicated BitTorrent proxies, like BTGuard or Torrent Privacy. Nevertheless, these services are not a guaranty of anonymity, since they are run by a single operator and can get compromised. Using the Tor network [1] for routing BitTorrent traffic is rather popular among downloaders. However, recent studies [2, 3] have proved that the Tor network is inadequate for anonymizing BitTorrent traffic, in addition to an official review[1] in the network website.

In the second case, several users formed part of anonymous file-sharing networks, like the Anomos network[2], or the I2P network [4]. Users within these networks preserve their anonymity while downloading among themselves, but the lack of recent or varied content make these networks unattractive for up-to-date file-sharing.

In this paper, we consider improving content availability in anonymous environments by taking into account public anonymous networks and their interactions with non-anonymous environments.

Our main goal is to study and develop the first model for internetwork communications: how can a user within an anonymous network access content in a non-anonymous one and remain anonymous? Our main contribution is to develop a fully operational interconnection model and test it with two current networks: the I2P network and the BitTorrent network.

This paper is organized as follows: Section 2 presents our measurements on the available content within the public BitTorrent environment and the I2P network file-sharing environment. Section 3 introduces our interconnection model, applied to both anonymous and non-anonymous environments. A fully implementation and further experiments of our model applied to the BitTorrent environment and the I2P network is presented in Section 4. Section 5 introduces our internetwork threat model and section 6 brings forward a set of important questions and ideas worth answering. Section 7 points out previous and current work on anonymous networks and anonymous file-sharing. Finally, Section 8 concludes this work.

[1] `http://blog.torproject.org/blog/bittorrent-over-tor-isnt-good-idea`
[2] `http://anomos.info/wp/category/anomos/`

2 Content Availability

It is fairly normal that popular content gets available first in public communities rather than in anonymous networks. We conduct a 30-days measurement to determine the rate of new content introduced per day in the public BitTorrent community, and in the I2P anonymous BitTorrent community.

We considered **Torrentz**, a major meta-search engine for BitTorrent content, which indexes torrent from various torrent sites, including *thepiratebay.org*, *mnova.eu* and *bitsnoop.com*. Regarding I2P, the **Postman** tracker was considered, which is the biggest BitTorrent tracker available within the I2P network.

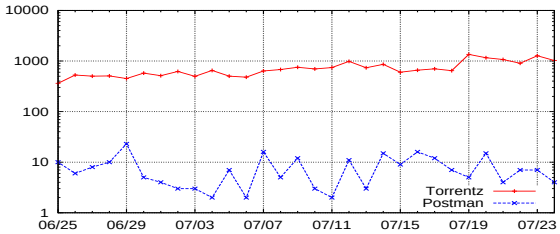


Fig. 1. New Content

For this measurement, we consider 4 main content categories: Movies, TV shows, Music and Games. Figure 1 presents the amount of new content introduced every day in the BitTorrent public community and in I2P. There are, in average, 720 new torrents in Torrentz and roughly 8 new torrents in the Postman tracker every day.

Trackers	Movies	TV Shows	Games	Music	Total
Torrentz	21.11%	47.45%	8.27%	23.15%	21648
Postman	48.30%	18.22%	2.96%	30.50 %	236

Table 1. 30-days measurement

Table 1 presents the amount of new content per category for every tracker, along with the total number of torrents measured, in which barely the 1% of the torrents in Torrentz are present in the anonymous I2P tracker. Moreover, Torrentz reports 19 million active torrents, while the Postman tracker reports around 12000 active torrents. These results point out the lack of content within the I2P network, and support our idea of automatically introducing new content to the I2P file-sharing community.

3 Internetwork Model

In this section we present our model to interconnect anonymous and public file-sharing networks.

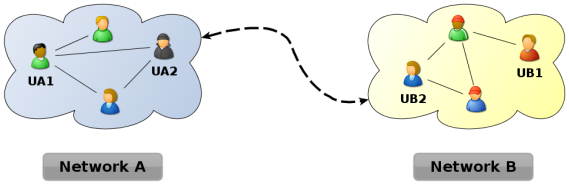


Fig. 2. Internetwork Scenario

We consider two networks A and B as shown in figure 2. Network A is a mixnet-based anonymous network, providing anonymity for its users while interacting among them: user UA_1 will be able to hide its location (normally its IP address) when contacting user UA_2.

Within the anonymous network, different users can have different anonymity settings, thus achieving more or less anonymity. Tunnels length in mixnet-based anonymous networks is one of the main factors affecting anonymity.

The network B is a non-anonymous network, in which users connect directly among them, without any IP hiding technique. If there is an interaction between the user UB_1 and the user UB_2, both parties will know each other IPs.

3.1 Design goal

Pfitzmann et al. [5] define **anonymity** of a subject as the impossibility to discriminate the subject among a set, called the *anonymity set*. **Unlinkability**, on the other hand, refers to the impossibility to link together two *items of interest* (e.g. a message, a user) from an attacker's point of view, for example determining that a message comes from a specific user. Our design goal is twofold. On the one hand, we aim to maintain the anonymity of a user UA_i while interacting with a user UB_j for the anonymity set:

$$\texttt{anonymity\_set} = \{\texttt{UA}_i, \texttt{UB}_j\} \ \forall i, j \tag{1}$$

On the other hand, we seek the unlinkability among a download for a given content and an anonymous user. It means that from an attacker's point of view, it must not be possible to determine which anonymous user in network A is downloading which content in network B.

3.2 Bridging model

For two users in different networks to interact, we need a *bridge*. A bridge is a component that allows internetwork traffic by actively participating in both networks. A bridge does not merely forward traffic, but can take decisions for improving the bridging, such as caching particular content, sharing content with other bridges, exchanging bridging information with other bridges or providing a tracker-like functionality among users.

Three different bridging approaches could be considered: on the one hand, a single bridge is a simple option, but it does not scale properly under network growth, especially if we think about different sets of users accessing different set

of content and the bridge managing all of these interactions at once. On the other hand, every user can perform its own bridging, which is an approach that clearly scales, but introduces new problems regarding a user anonymity.

The third option, and the one we consider to be the most proper and accurate solution for our internetwork model, is an overlay of bridges. Figure 3 presents our model for anonymous internetwork file-sharing. It is divided into three components, the anonymous network, the non-anonymous network and the set of bridges, and presents the following properties:

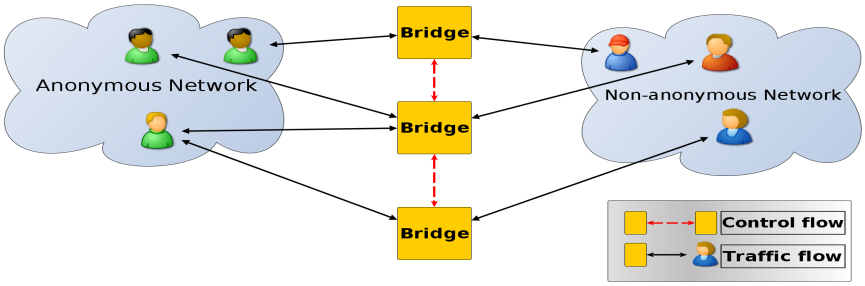


Fig. 3. Internetwork model

File-sharing protocol. Users in both networks use the same file-sharing protocol to communicate, therefore a bridge only speaks a single file-sharing protocol.

Bridges deployment. Bridges are dynamic and can be started on demand.

Interaction point: A bridge is the only visible component for both networks: users will only see these bridges (probably a sub-set of the total bridges) as the only connection point with the other network.

Number of bridges. The minimum number of bridges required is defined in (2), where `UA` is the number of users in network A to interconnect, `MUA` is the maximum number of users a bridge can manage in network A, `UB` is the number of users in network B to interconnect and `MUB` is the maximum number of users a bridge can manage in network B.

$$\texttt{nb\_bridges= MAX(} \lceil \texttt{\#UA/\#MUA} \rceil, \lceil \texttt{\#UB/\#MUB} \rceil \texttt{)} \tag{2}$$

Taking as an example figure 2 and assuming a single bridge can manage two anonymous users and one non-anonymous user at a time, we need $\texttt{nb\_bridges = MAX(}\lceil \texttt{3/2} \rceil, \lceil \texttt{3/1} \rceil\texttt{) = MAX (2,3) = 3}$ bridges to manage a full interconnection. The maximum number of users a bridge can manage depends on the network itself: we probably need less bridges in a file-sharing network than in a real-time video network with the same number of users, since the *throughput* in a file-sharing environment is less critical and bridges can manage more users.

A bridge has an available bandwidth, and we can think on adding extra bridges to an existing interconnection so as to increase the total available bandwidth. If `b_bw` is the bridge's available bandwidth, the total available bandwidth with `X` bridges will be $\texttt{BW(b\_bw,X) = b\_bw} * \texttt{X}$. Thus further bridges can be

added to increase the total available bandwidth within an interconnection, an interesting characteristic in file-sharing networks.

3.3 Bridge anonymity

Bridges have their anonymity decreased to zero, being that they connect directly to the public network. Nevertheless, this property does not affect the anonymity nor the unlinkability provided by our design.

On the one hand, anonymous peers use the anonymity chain provided by the anonymous network to connect to our bridges, therefore the anonymity of these anonymous peers is maintained within the anonymity set. Even by placing a malicious bridge, an attacker can not de-anonymize an anonymous user.

On the other hand, from an attacker's point of view, it can only link a download for a given content with a particular bridge and not further, thus achieving the required unlinkability between anonymous users and public content.

4 Interconnecting I2P and public BitTorrent Swarms

In this section we apply our model to an existing environment and evaluate its performance.

4.1 The I2P network

We focus on the I2P network, a low-latency network layer which provides anonymity for identity-sensitive applications. This network provides a set of built-in applications, such as P2P clients, an email client and an IRC client. Within I2P, users can, for example, use a BitTorrent application to share content, as in any normal BitTorrent environment, forming normal BitTorrent swarms, called *I2P swarms*. I2P users can not access non-I2P content and vice-versa.

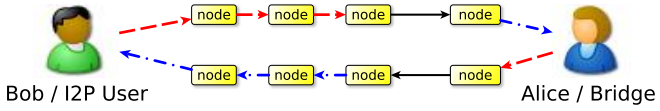


Fig. 4. Bridge incoming/outgoing tunnels

It means that if the latest Ubuntu distribution, for example, wants to be shared within I2P through a torrent, an I2P user needs to 1) download the file through a non-anonymous source, like with a simple BitTorrent client, 2) create the torrent file for that file and publish it in the I2P torrent tracker and 3) host and share this file throughout the time it will take to spread the file to other I2P users.

This procedure to add new content in the I2P network requires a dedicated user to bootstrap the content and host it throughout several downloads for remote users. It has several disadvantages:

– **Active participation:** A dedicated I2P user is required to perform every previously mentioned step to add new content.
– **Exposed identity:** This dedicated user needs to download the file from another source, probably a non-anonymous one, exposing its identity.
– **Reduced sources:** During the initial downloads, this unique peer will be the only available source of the content. If this user goes offline, the downloads will stall.

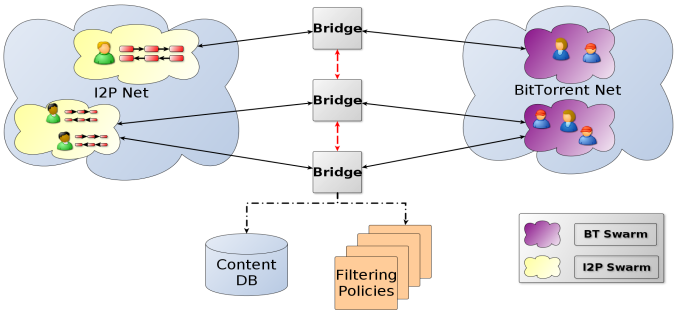


Fig. 5. Internetwork model applied to I2P and BitTorrent

I2P is a mixnet-based network, which uses a series of *tunnels* to route traffic within the network. These tunnels are formed by the I2P users and allow an indirect communication between two users. Figure 4 presents a simple tunnel-oriented I2P communication, in which Bob defines a 2-hop tunnel for its incoming and outgoing communications, while Alice defines a 0-hop tunnel. The anonymity of a user is based on the length of its tunnels: choosing the right length of a tunnel is a tradeoff between speed and anonymity that needs to be taken into consideration every time a user runs the I2P software.

An I2P user has a *destination* associated, which replaces the normal IP address within the network: an I2P user receives data to/from a destination and no longer to an IP address.

4.2 Bringing together I2P and BitTorrent swarms

We consider the normal BitTorrent network as the non-anonymous network, which is formed by several swarms, sharing thousand of different files at any moment. Our model will be applied to both these networks so as to allow anonymous I2P users to join any normal BitTorrent swarm while maintaining their anonymity.

Figure 5 presents a graphical view of our architecture with these two networks. I2P swarms will be able to connect to BitTorrent swarm thus forming an unique swarm, composed with both types of users, anonymous and non-anonymous. On the one hand, our bridge has 0-hop inbound/outbound tunnel to increase its throughput. Since a bridge directly connects to BitTorrent users

with a non-anonymous connection, there is no need to maintain its anonymity within the I2P network

On the other hand, I2P users will keep connecting through their self-defined inbound/outbound tunnels, thus preserving their anonymity intact.

Based on our preliminary tests, we consider a value of 40 connections between a bridge and I2P/BitTorrent users to begin with, despite current BitTorrent clients easily manage above 80 connections per swarm. Current work includes testing a bridge with extra I2P/BitTorrent user connections, and determining how these further connections impact the performance.

Therefore the number of bridges required for interconnecting a given swarm is now $nb\_bridges(UA, UB) = MAX(UA/40, UB/40)$.

4.3 Copyrighted content filtering

Since we allow P2P traffic to move between networks, we consider it is highly important to take into account copyrighted content, which can not be freely and legally distributed.

We put in place a database, which holds the *infohash* (the BitTorrent identifier for a given content) for different content, organized in categories, such as movies, music, games, TV shows, software, etc. The content can be a copyrighted book not authorized for free distribution, or it can a content suitable for free distribution, such as the HD space mission footage from NASA. An simplify extract of the non-copyrighted content list from our database can be downloaded from `http://i2pstats.loria.fr/bridging/`

Filtering policies are used to discriminate which content to bridge: a bridge operator might want to avoid bridging the space missions from NASA, or just want to bridge non-copyrighted music. Thus, every one of our bridges periodically queries the database for new content, and based on its own filtering policies, it accepts or denies bridging particular contents.

Currently, we only provide filtering policies to filter content by category, however current work is focused on developing further complex conditions for content filtering. Examples of filtering policies can be found in the previously mentioned website.

Figure 5 shows only one bridge accessing the content database for simplification. However, every bridge accesses this database and holds its own filtering policies.

4.4 Bridge operation improvements

We improve our bridge over a regular BitTorrent client in two ways.

Full piece request. A BitTorrent peer will normally request different blocks for the same piece (A BitTorrent piece is divided into blocks), before choosing the next piece to download. When a bridge receives a block request from the I2P side, it downloads the entire piece from the BitTorrent side, thus following requests will be answered with no delay from the bridge.

Piece caching. A bridge operator might choose to perform *piece caching* to improve the time response for piece requests coming from the I2P network. We currently define two possible caching policies. The first policy states that every requested piece should be stored for a small period of time, normally 60 seconds. The second policy specifies to take into consideration the *rarest pieces*, and only store them in the cache. The BitTorrent's rarest first algorithm [6] states that a peer will choose to download those pieces which the fewest of their connected peers have. We consider the same set of pieces, and keep in the cache only those pieces for an interval of time, normally 120 seconds.

4.5 I2P User-Bridge interaction

Figure 6 presents every step of an interaction between an I2P user and a bridge, which includes the following steps:

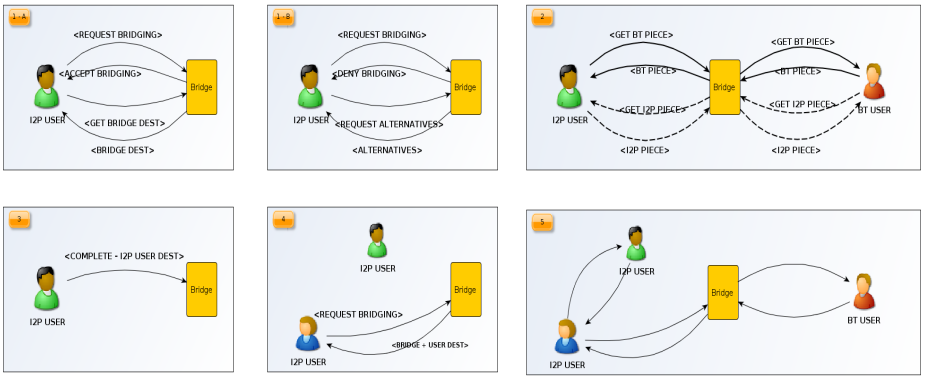


Fig. 6. Interaction between an I2P User and a Bridge

- `Step 1a:` An I2P user requests bridging and if accepted, it requests the bridge I2P destination.
- `Step 1b:` In case the bridge denies the request, the I2P user requests alternative bridges and executes step `1a` once again.
- `Step 2:` The bridge connects to BitTorrent users and the download starts. Both I2P and BitTorrent users can request pieces.
- `Step 3:` Once the I2P user completes the download, it announces its new *Seeder* status to the bridge.
- `Step 4:` When a new I2P user requests bridging, the bridge will return its destination and the one of the other I2P peer.
- `Step 5:` The new I2P user connects to both the bridge and the existing I2P user and starts the download.

Step 4 presents the tracker functionality of the bridge, which keeps track of the I2P users sharing a given content. When an I2P user contacts the bridge, it

will return not only the destination of its built-in I2PSnark client, but also the destinations of other I2P users (if any) already sharing that content. This way, I2P users will form an I2P swarm and be able to share pieces of the content among themselves.

4.6 Downloads measurements: Single bridge

We chose the top 20 torrents from the Postman I2P tracker regarding swarm size within the I2P network (an average of 15 seeds in every swarm), and measured the download rates achieved during the downloads. Since the swarm speed is computed as the sum of every single peer connection, we considered both the total download rate of the swarm and of the fastest peer.

Figure 7 presents the results for normal I2P downloads. The fastest peer in every swarm presents an average download rate of 9 KBps, and the swarms a total download rate of 33 KBps. However, and in few occasions, we were able to achieve a 70 KBps download rate in the swarm (with 23 Kbps for the fastest peer) during a 5-minute period. If the I2P swarm is formed by several fast peers, with fairly fast bandwidth settings, it is possible to achieve higher swarm download rates, since the total speed is calculated as the sum of every peer in the swarm.

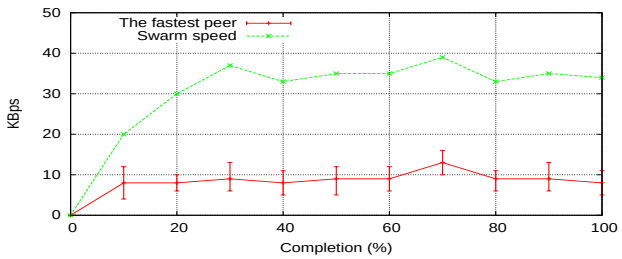


Fig. 7. I2P download rates

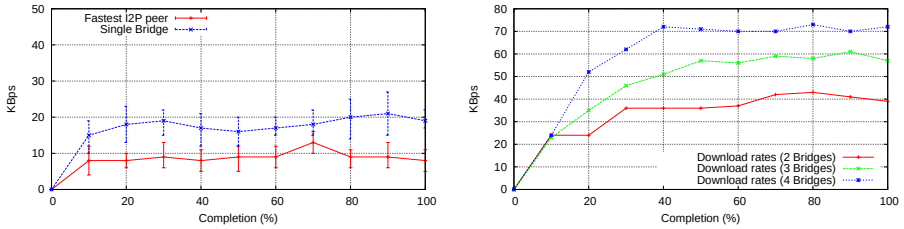
Figure 8(a) presents the download rate achieved with one of our bridges along with the fastest measured I2P peer. In this case, the measurement configuration is, on the one side, a single bridge joining a BitTorrent swarm with one peer, and one the other side, an I2PSnark client, downloading a 10 MB BitTorrent file through our bridge.

After 26 downloads of the file, we measured an average download rate of 17 KBps from the bridge. During our measurements, in some occasions, we had an average download rates of 37 KBps, with peaks of 80 KBps.

We compare the fastest peer in a I2P swarm and our bridge: as expected and due to the 0-hop inbound/outbound tunnels in the bridge, the download rate achieved is higher, rising at almost the double of a normal I2P peer.

4.7 Downloads measurements: Several bridges

We conducted a second measurement, in which we increased the number of bridges within an I2P swarm. We took the same configuration as before, which includes one I2P peer, one BitTorrent peer and now a set of bridges.



(a) Single bridge download rate (b) Several bridges download rates

Fig. 8. Leasesets results

Figure 8(b) presents the achieved download rates with 2, 3 and 4 bridges, all bridging at the same time. As mentioned in section 3, the available download rate is proportional to the number of bridges present in the swarm. If a bridge normally achieved around 17 KBps, with 2 bridges we can have 34 KBps, with 3 bridges approximately 50 KBps, and so on.

Assuming we have a large set of available bridges, we can increase the minimum require bridges for an interconnection, define by equation (1), aiming only to increase swarm's speed.

4.8 Public bridge implementation

Our internetwork model is fully implemented and freely usable. A slightly modified I2PSnark client can be downloaded from `http://i2pstats.loria.fr/bridging/`, which includes the functionality to contact our bridges in case the user decides to download a non-I2P torrent. Otherwise, this modified client behaves exactly as any normal I2PSnark client.

Our current filtering policies only allow bridging a set of NASA space mission videos, which have been released into the public domain. The torrents for these contents are available in the previous mentioned website, or can be downloaded from `http://www.mininova.org/`.

Therefore, any I2P user can replace the original I2PSnark file with our modified version, load any of the provided torrents and initiate a fully anonymous download for public BitTorrent content.

A downloadable version of our bridge will be soon available in our website for any user willing to test it and improve the overlay of existing bridges.

5 Threat Model

This section introduces our threat model, which is mainly focused on a malicious bridge operator.

5.1 Malicious bridge

Being our bridge implementation freely downloadable, it is possible to encounter a malicious bridge, aiming to perform either passive or active attacks, such as monitoring the downloads or de-anonymizing I2P users.

On the one hand, I2P users utilize tunnel-based channels to reach a bridge, as well as with any other remote I2P user. The anonymity achieved by an I2P user depends on its own inbound/outbound tunnels, and not on the remote I2P user tunnels, as mentioned in section 4.1. Therefore, a malicious bridge, who only controls its own tunnels, will not be able, under any circumstances, to de-anonymize an anonymous user.

On the other hand, we consider that maintaining a list of the current bridged content does not affect an I2P user, since a bridge will not be able to link a specific I2P user with a specific download.

5.2 Monitoring public BitTorrent Swarms

Since a bridge forms part of a public BitTorrent swarm, it can link any download with any BitTorrent peer involved in that swarm. However, any normal BitTorrent user can perform this task by joining any given swarm, and keeping track of its connected peers, an extremely simple technique for monitoring. Therefore, a bridge does not introduce new security threats for BitTorrent users.

6 Discussion

In this section we answer a series of open questions we have considered within our work.

Can a bridge operator be exposed to illegal downloads? We enable a bridge operator to filter which content to bridge through *filtering policies*, which gives the bridge operator a total control regarding the allowed content. However, if no filtering policies are specified, a bridge will forward any type of content, which might lead to an illegal download. It is always possible for the bridge operator to argue *plausible deniability*, indicating that the P2P traffic is generated from third parties. An alternative option is to anonymize the outgoing traffic produced by a bridge by means of another anonymous network layer or proxy, which reduces throughput, but increases the anonymity of the bridge operator.

It is important to notice that merely anonymizing BitTorrent traffic with an anonymous layer, such as BitTorrent traffic over Tor, is not enough to achieve an anonymous download. We provide both anonymous content indexation and anonymous content distribution, whereas the previous approach only provides anonymous content distribution.

Why is the I2P network more suitable for file-sharing than other networks? In this work we selected the I2P network based on its features: a strong anonymity for the end user, a fairly complete threat model and its built-in

features, such as email, IRC and file-sharing applications. The I2P network has seen a considerable increase in its user base during the last months, reaching approximately 15000 users, which clearly indicates that the network is growing, making it an interesting target for research.

How is our bridge different from an open proxy? A proxy is one of the options that users have to hide their real IP addresses, for achieve anonymous file-sharing, for example. A user can route *any kind* of traffic through this proxy, not only BitTorrent traffic. However, our bridge is enhanced for BitTorrent traffic, incorporating features such as *piece caching* and *tracking capabilities*, allowing a complete anonymous network to access public content. The alternative would be for every user to use an open proxy to route their traffic, leading to a set of Ad hoc independent connections rather than a smart interconnection of networks.

7 Related work

There have been several efforts for achieving anonymous file-sharing, from dedicated anonymous networks, to proxies and network layers.

Anonymous networks, such as Freenet, I2P [4] or GNUnet, allow users to anonymously share content within the network limits. These networks are not usually proxies for the World Wide Web, and the only available content within these networks is that one that has been previously inserted in them.

Freenet [7] is a peer-to-peer decentralized and distributed data storage, mainly designed as a censorship-resistant platform. The network can be used in *darknet mode* or *opennet mode*, however in both cases only previously added content to the Freenet network can be accessed.

GNUnet [8] is platform for anonymouos file-sharing, with support for a *darknet mode* operation as well. As with Freenet, GNUnet does not have any proxy for the regular Internet, therefore internetwork file-sharing can not be performed.

Public proxies such as JAP, or networks layers such as the Tor network route a user's traffic between different intermediate nodes, thus hiding the IP of the user from the destination of the traffic.

JAP [9] uses a sequence of anonymization proxies or also called a *mix cascade* to provide pseudoanonymous web browsing. A user can select among a set of fixed cascades to communicate with another user, which basically groups users into large anonymity sets.

The Tor network [1] provides a circuit-based anonymous communication service based on the onion routing [10]. It has been widely adopted and it is one of the principal options for pseudoanonymous web browsing.

A user seeking anonymous file-sharing can use these types of system to hide its real IP, while accessing content in the World Wide Web. However a major problem with these low-latency anonymous systems is that they are vulnerable to traffic confirmation or *end-to-end correlation* [3], which can affect the anonymity achieved while downloading.

Internetwork file-sharing, on the contrary, has not been widely explored.

OneSwarm [11] proposes a privacy-centered file-sharing protocol, allowing a user to specify the level of trust with other peers in the OneSwarm network, as well as with the data a user keeps: this data can be publicly shared, anonymously shared or shared with some restrictions.

However, OneSwarm does not allow active interconnection between two networks. The user has to bootstrap new content in the network and make it available for the rest of the users, exposing itself to a non-anonymous download. Additionally, Prusty et al. [12] demonstrate that OneSwarm's vulnerability to traffic analysis is greater than previous reported in [11].

8 Conclusion

We tackle the problem of improving content availability in anonymous environments, by interconnecting an anonymous network with public Swarms. By bringing together I2P Swarms along with BitTorrent Swarms, we enable a secure content indexation along with an anonymous content access, as opposed to file-sharing over the Tor network, which only enables anonymous content access.

With our model, I2P users are now able to access non-anonymous BitTorrent content without compromising their anonymity. We have shown that the download rate with a single bridge is enough to download a 700 MB file within 10 hours, which is fairly acceptable considering it is an anonymous download. Additional bridges can be placed to increase the overall bandwidth of a swarm.

On the one hand, I2P offers both TCP and UDP transport protocols, which allows *all* BitTorrent traffic to be routed through the network, in contrast with Tor. On the other hand, a user can achieve a few hundred KB/s when routing its BitTorrent traffic over Tor, which is significantly faster than in our case. Bandwidth rates within I2P depends on the number of fast peers, and as the network gains more users, it will offer higher bandwidth rates.

Future work consists in improving our content checking mechanism, including RSS feed-like approaches, to automatically update our content database with new and current content.

Additional future work consists on evaluating bridge interconnection towards three main directions:

Bridging network awarenesses. When an anonymous user requests bridging for a given content, a bridge can check whether other bridges are already dealing with this content and forward the request.

Pieces sharing. Since several bridges can be present in a given swarm, they can share different pieces of a same content depending on the requests they receive. If a user requests a piece which is no longer available in a given bridge, the bridge can obtain it from another bridge directly and forward it to the user.

Handoffs. When overloaded, a bridge can request another one to take over an internetwork connection and reduce its load.

Acknowledgment: We thank to Tarang Chugh for his valuable reviews and his work during the bridge implementation process.

References

1. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, USENIX '04, San Diego, California, USA, August 2004. USENIX.
2. P. Manils, A. Chaabane, S. le Blond, M.A. Kaafar, C. Castelluccia, A. Legout, and W. Dabbous. Compromising Tor Anonymity Exploiting P2P Information Leakage. In *Proceedings of the 3rd Hot Topics in Privacy Enhancing Technologies*, HotPETs '10, Berlin, Germany, July 2010. IEEE Communications Society.
3. Stevens Le Blond, Pere Manils, Abdelberi Chaabane, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One bad apple spoils the bunch: exploiting P2P applications to trace and profile Tor users. In *Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats*, LEET '11, Berkeley, CA, USA, March 2011. USENIX Association.
4. I2P. The I2P network. `http://www.i2p2.de/`.
5. Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Proccedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, Berkeley, California, USA, July 2001. Springer-Verlag New York, Inc.
6. Bram Cohen. Incentives Build Robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, Berkely, California, USA, June 2003.
7. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, Berkeley, California, USA, July 2001. Springer-Verlag New York, Inc.
8. Krista Bennett, Christian Grothoff, Tzvetan Horozov, Ioana Patrascu, and Tiberiu Stef. GNUnet - A truly anonymous networking infrastructure. In *Proceedings Proc. Privacy Enhancing Technologies Workshop (PET)*, PET '02, San Francisco, CA, USA, April 2002. Springer.
9. Oliver Berthold, Hannes Federrath, and Stefan Kpsell. Web MIXes: A system for anonymous and unobservable Internet access. In *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, Berkeley, California, USA, July 2001. Springer-Verlag New York, Inc.
10. D. Goldschlag, M. Reed, and P. Syverson. Hiding routing information. In *Proceedings of the 1st International Workshop on Information Hiding*, IH '96, Cambridge, UK, May 1996. Springer.
11. Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas Anderson. Privacy-preserving P2P data sharing with OneSwarm. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, New Delhi, India, August 2010. ACM.
12. Swagatika Prusty, Brian Neil Levine, and Marc Liberatore. Forensic investigation of the OneSwarm anonymous filesharing system. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, New York, NY, USA, October 2011. ACM.